

BAB 3: IMAGE TRANSFORMATION

1. Tujuan

- Mahasiswa dapat melakukan teknik transformasi pada citra digital
-

2. Dasar Teori

Transformasi geometri pada pengolahan citra digital adalah teknik matematika yang digunakan untuk mengubah posisi, orientasi, skala, dan bentuk gambar. Transformasi ini diterapkan pada setiap piksel dalam citra dan dapat mengubah ukuran, bentuk, dan orientasi gambar.

Jenis-jenis transformasi geometri yang umum digunakan dalam pengolahan citra digital antara lain sebagai berikut.

1. Translasi

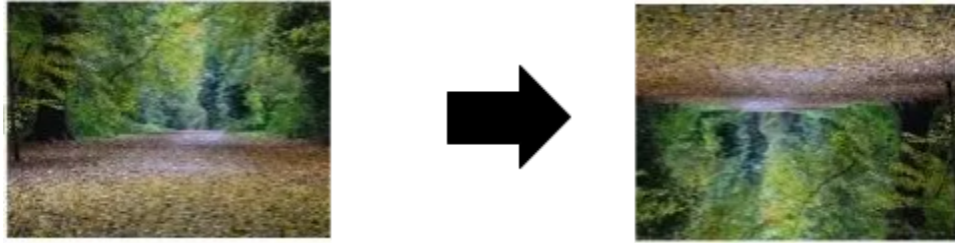
Translasi adalah transformasi yang digunakan untuk memindahkan gambar ke posisi yang berbeda tanpa mengubah ukuran atau orientasinya. Translasi dilakukan dengan menggeser semua piksel gambar sejauh vektor tertentu pada sumbu x dan y. Hasil dari translasi adalah gambar yang sama persis dengan gambar asli, tetapi terletak pada posisi yang berbeda.



Gambar 1. Contoh translasi pada citra (sumber www.cronj.com)

2. Rotasi

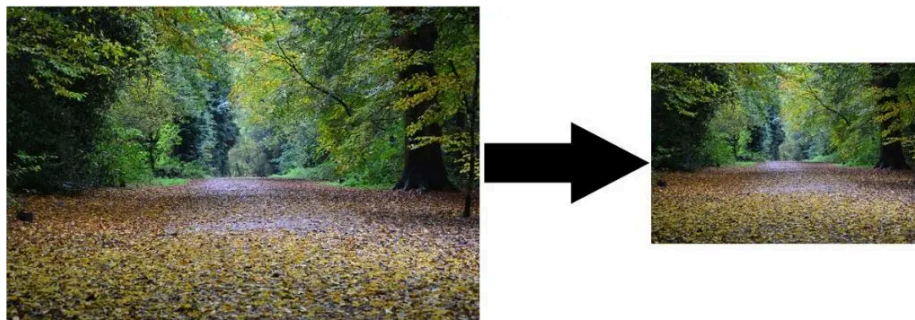
Rotasi adalah transformasi yang digunakan untuk memutar gambar sebesar sudut tertentu dalam arah yang berlawanan dengan arah jarum jam atau searah dengan arah jarum jam. Rotasi dapat dilakukan dengan menggunakan matriks rotasi, dan titik pusat rotasi dapat ditemukan dengan mencari titik pusat dari gambar. Hasil dari rotasi adalah gambar yang terlihat seperti gambar asli, tetapi dalam orientasi yang berbeda.



Gambar 2. Contoh rotasi pada citra (sumber www.cronj.com)

3. Skala

Skala adalah transformasi yang digunakan untuk mengubah ukuran gambar secara proporsional. Skala dapat ditingkatkan atau dikurangi. Skala dapat diterapkan pada sumbu x dan y secara terpisah atau dapat diterapkan pada kedua sumbu secara bersamaan. Skala dapat dilakukan dengan menggunakan matriks skala. Hasil dari skala adalah gambar yang lebih kecil atau lebih besar dari gambar asli.



Gambar 3. Contoh skala pada citra (sumber www.cronj.com)

3. Alat dan Bahan

- **Komputer** (Windows, macOS, atau Linux).
 - **Browser Modern** (misalnya Google Chrome, Mozilla Firefox, atau Microsoft Edge).
 - **Text Editor/IDE:**
 - Visual Studio Code
 - Atau alternatif: CodePen.io (untuk eksperimen langsung secara online).
 - **File Source Code:**
 - `index.html`
 - `apps.js`
 - **Gambar Digital** (JPEG, PNG, dll) sebagai sampel input.
-

4. Langkah Kerja

Langkah 1: Persiapan Proyek

1. Buat sebuah folder proyek baru.
2. Di dalam folder tersebut, buat dua file: `index.html` dan `apps.js`.

Langkah 2: Menulis Kode HTML

Buat file `index.html` dengan konten berikut:

```
1. <!DOCTYPE html>
2. <html lang="id">
3. <head>
4.   <meta charset="UTF-8">
5.   <title>Aplikasi Pemrosesan Citra Digital Sederhana Part 2</title>
6.   <style>
7.     body {
8.       font-family: sans-serif;
9.       margin: 20px;
10.    }
11.    #canvas {
12.      border: 1px solid #ccc;
13.      margin-top: 10px;
14.    }
15.    .channel-canvas {
16.      border: 1px solid #ccc;
17.      margin: 10px;
18.    }
19.  </style>
20.</head>
21.<body>
22.  <h1>Aplikasi Pemrosesan Citra Digital Part 2</h1>
23.  <input type="file" id="upload" accept="image/*">
24.  <br><br>
25.  <label for="operation">Pilih operasi:</label>
26.  <select id="operation">
27.    <!-- ▼ Tambahan transformasi geometri ▼ -->
28.    <option disabled>————</option>
29.    <option value="translate">Translasi (geser)</option>
30.    <option value="scale">Skala</option>
31.    <option value="rotate">Rotasi</option>
32.    <option value="flipH">Balik horizontal</option>
33.    <option value="flipV">Balik vertikal</option>
34.  </select>
35.  <button id="process">Proses</button>
36.  <br><br>
37.  <canvas id="canvas"></canvas>
38.  <!-- Kontainer untuk hasil channel terpisah -->
39.  <div id="channels"></div>
40.
41.  <script src="apps.js"></script>
42.</body>
43.</html>
```

Langkah 3: Menulis Kode JavaScript

Buat dan atau buka file **apps.js** dan bagi penulisan kode menjadi beberapa bagian berikut.

3.1. Kerangka JavaScript

Buat struktur dasar untuk memuat gambar, mengambil data dari canvas, dan memanggil fungsi sesuai pilihan operasi.

Pertama, tambahkan fungsi DOMContentLoaded. Blok fungsi ini akan berfungsi ketika element pada html selesai di load.

```
document.addEventListener('DOMContentLoaded', function() {  
  
    // Load gambar dan tampilkan pada canvas  
    upload.addEventListener('change', function(e) {  
    });  
  
    // Proses gambar sesuai pilihan operasi  
    processButton.addEventListener('click', function() {  
    });  
  
    // Fungsi pendukung akan ditambahkan di bawah ini.  
    // ...  
});
```

Kedua, tambahkan variabel untuk dipasangkan dengan elemen HTML berdasarkan ID elemen.

```
document.addEventListener('DOMContentLoaded', function() {  
  
    const upload = document.getElementById('upload');  
    const canvas = document.getElementById('canvas');  
    const ctx = canvas.getContext('2d');  
    const processButton = document.getElementById('process');  
    const operationSelect = document.getElementById('operation');  
    let originalImageData = null;  
  
    // Load gambar dan tampilkan pada canvas dan seterusnya  
});
```

Ketiga, tambahkan fungsi didalam 'DOMContentLoaded' untuk menangani proses unggah gambar

```
// Load gambar dan tampilkan pada canvas
upload.addEventListener('change', function(e) {
  const file = e.target.files[0];
  if (!file) return;
  const reader = new FileReader();
  reader.onload = function(event) {
    const img = new Image();
    img.onload = function() {
      canvas.width = img.width;
      canvas.height = img.height;
      ctx.drawImage(img, 0, 0);
      originalImageData = ctx.getImageData(0, 0, canvas.width, canvas.height);
    }
    img.src = event.target.result;
  }
  reader.readAsDataURL(file);
});
```

Keempat, tambahkan fungsi berikutnya didalam 'DOMContentLoaded' untuk menangani tombol diklik

```
// Proses gambar sesuai pilihan operasi
processButton.addEventListener('click', function() {
  if (!originalImageData) return;
  const op = operationSelect.value;

  switch(op) {
    case 'translate':
    case 'scale':
    case 'rotate':
    case 'flipH':
    case 'flipV':
      geometricTransform(op);
      break;
    default:
      return;
  }
});
```

3.2. Fungsi Transformasi Geometri

Fungsi untuk melakukan transformasi citra

```
1. function geometricTransform (type) {
2.   const offCanvas = document.createElement('canvas');
3.   offCanvas.width = canvas.width;
4.   offCanvas.height = canvas.height;
5.   const offCtx = offCanvas.getContext('2d');
6.
7.   switch (type) {
8.     case 'translate': {
9.       const dx = parseFloat(prompt('Geser horizontal (dx piksel):', '50')) ||
0;
```

```

10.         const dy = parseFloat(prompt('Geser vertikal (dy piksel):', '50'))
    || 0;
11.         offCtx.translate(dx, dy);
12.         break;
13.     }
14.     case 'scale': {
15.         const sx = parseFloat(prompt('Skala horizontal (sx):', '1.5')) || 1;
16.         const sy = parseFloat(prompt('Skala vertikal (sy):', '1.5')) || 1;
17.         offCtx.scale(sx, sy);
18.         break;
19.     }
20.     case 'rotate': {
21.         const deg = parseFloat(prompt('Sudut rotasi (derajat):', '45')) ||
0;
22.         const rad = deg * Math.PI / 180;
23.         // rotasi di pusat gambar
24.         offCtx.translate(canvas.width/2, canvas.height/2);
25.         offCtx.rotate(rad);
26.         offCtx.translate(-canvas.width/2, -canvas.height/2);
27.         break;
28.     }
29.     case 'flipH':
30.         offCtx.translate(canvas.width, 0);
31.         offCtx.scale(-1, 1);
32.         break;
33.     case 'flipV':
34.         offCtx.translate(0, canvas.height);
35.         offCtx.scale(1, -1);
36.         break;
37.     }
38.
39.     /* gambar ulang ke kanvas baru, lalu salin kembali */
40.     offCtx.drawImage(canvas, 0, 0);
41.     ctx.clearRect(0, 0, canvas.width, canvas.height);
42.     ctx.drawImage(offCanvas, 0, 0);
43. }

```

Catatan: Pastikan seluruh fungsi pendukung (poin 3.2 sampai 3.6) ditempatkan di dalam kerangka event listener **DOMContentLoaded** atau diimpor secara terpisah sehingga dapat diakses oleh kode utama.

Langkah 4: Pengujian Aplikasi

1. Buka file **index.html** di browser menggunakan Visual Studio Code (dengan Live Server) atau unggah ke CodePen.io.
2. Pilih sebuah gambar melalui input file.
3. Pilih salah satu operasi dari dropdown (misalnya: translasi, skala, rotasi, balik).
4. Klik tombol **Proses** untuk melihat hasilnya pada canvas

Langkah 5: Eksperimen dan Analisis

1. Uji setiap filter dan perhatikan perbedaan hasil dari masing-masing operasi.

2. Cobalah mengubah nilai-nilai parameter (misalnya nilai x atau y saat translasi) untuk mengamati perubahan hasil citra.
-

5. Tugas

1. **Analisis Teoritis:**

- Jelaskan perbedaan mendasar antara translasi dan rotasi

2. **Peningkatan Antarmuka:**

- Rancang ulang tampilan antarmuka agar lebih interaktif dan informatif dengan CSS dan JavaScript.
- Tambahkan fitur untuk mengunduh hasil citra yang telah diproses.

3. **Refleksi Praktikum:**

- Buat laporan singkat mengenai kesulitan yang ditemui selama praktikum beserta solusi yang dicoba.
 - Sertakan analisis perbandingan hasil tiap fungsi
-